Project 2: Milestone

Bartolo Medical Sales Inc. Customer Relationship Management Database

Michelle Bartolo

Software Development- CMPT 220L

Professor Pablo Rivas

6 April 2016

Abstract

This paper details the motivations, process, and methods to create a database for Bartolo Medical Sales Inc. I wanted to create a database in order to organize the data for my father's business, Bartolo Medical Sales, in order to help him organize information about the companies he represents and customers he sells to. I utilized the Netbeans, which is an open-source integrated development environment (IDE) for developing with Java and other programming languages, to create the database that connects to a Java Darby or Java DB database. For my project, I created a database with a user interface that shows information for customers that buy from Bartolo Medical Sales. For each customer entered into the database, the name of the company, the name of the contact, their phone number, email, and address are stored into a table.

Introduction

Bartolo Medical Sales is a small Hospital and Health Care company founded in January 1991 by Domenic Bartolo, a graduate of Bernard Baruch College. The company sells various medical products in the Metropolitan New York area to a diversified medical dealer network. Bartolo Medical Sales evaluates the needs of prospective clients, drafts proposals, and makes formal presentations and demonstrations to market products and close sales. I was motivated to complete this project of creating a Customer Relationship Management (CRM) Program for the company in order to improve the sales of Bartolo Medical Sales and allow it to become increasingly successful. It is extremely beneficial for a business to use some sort of CRM program because it enables individuals to keep track of communications with customers in an organized manner. Unfortunately, these programs cost hundreds and sometimes thousands of dollars and do not always have all of the features wanted and needed. This database stores information, such as email, phone number, and address, of clients so that Bartolo Medical Sales Inc. can continue to effectively sell medical equipment without having to pay thousands of dollars purchasing programs.

Detailed System Description

Java uses JDBC or Java Database Connectivity to connect to databases. There's a JDBC API, which is the programming part, and a JDBC Driver Manager, which programs use to connect to the database. JDBC allows a connection to various types of databases, such as Oracle, MySQL, etc. I used the inbuilt database that you get with the Java/NetBeans software. The database is called Java DB, which is a version of Apache Derby. It runs on a virtual server, which stops and starts from within NetBeans. The first thing I did was start a server at JavaDB, which allowed me to create a database. I then created a table for the database with columns called “ID”, “Company_Name,” “Contact_Name”, “Phone_Number”, “Email”, and “Address.” The ID column holds a unique identifying number, which identifies a specific row in the table. It is a primary key, which means it cannot be null and it is an integer. For the rest of the columns, I set the datatype as varchar. I then added data to this table by using sql commands to add new rows.

After I created the database, I started a new Java Application project and called the package database_console, and the main class DBConnect. To connect to a database, I had to add imports for connection, driver manager and sql exception. To set up a connection to a database, I added a line of code that takes my username, password, and host location of the database. I put that into a try...catch statement to make sure that the SQLException error doesn’t occur and added a client driver which allows it to be connected successfully. Using various commands that connect the columns of the table I created to the code, I printed out the rows of the table.

After that, I created a user interface for the database, which allows the users to see their records in a more organized manner. It lets the user press buttons to scroll backwards and forwards through their data, go to the first and last records, add new records, update a record, delete a record, save the record they want to add or cancel adding a new record if they change their mind. To do this I made a JFrame form which allowed me to create a GUI application and connect it to my database using java.

To connect the "Next" button I created with the GUI application to my code and make it functional, I created an action performed method which connects with the variable name of the button. I used an "if, else" statement check to check if there is a next record to move to, rs.next() and if there is a next record, I made it display it in the Text Fields. The if else statement is wrapped in a try … catch block, to prevent a SQLException error. If there's no next record, an alert pops up saying "End of file."

To connect the "Previous" button that I created with the GUI application to my code and to make it functional, I created an action performed method which connects with the variable name of the button. To go backwards through the records, I utilized the same commands as the next button, but instead of rs.next() I used the rs.previous() function.

To connect the "First" button I created with the GUI application to my code and make it functional, I created an action performed method which connects with the variable name of the button. I did not need an "if, else" statement because it is not necessary to check if there is a next or previous record since we are just going to the first record. The only thing I needed to do was move to the first record with rs.First, then display the first record in the Text Fields.

To move to the last record using the last record button, I used the same commands as I did for the first record button, but I used rs.Last instead of rs.First.